

BÀI 6: LẬP TRÌNH CƠ SỞ DỮ LIỆU

Mục tiêu:

Trong bài này, Anh/Chị cần đạt được mục tiêu sau:

Thực hành kỹ năng xây dựng các giao diện ứng dụng thông qua sử dụng các điều khiển nội tại và điều khiển dữ liệu thường dùng. Viết lệnh xử lý các sự kiện thường gặp trong một ứng dụng hướng sự kiện.

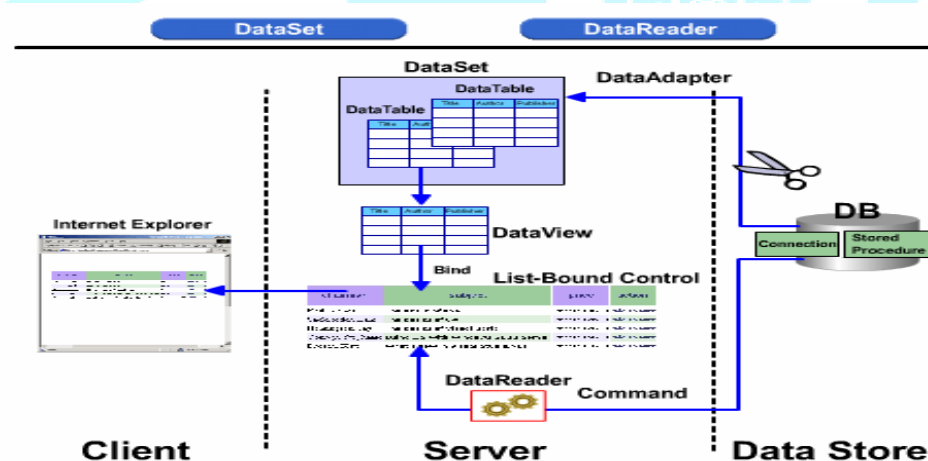
Nội dung:

Bài 6: LẬP TRÌNH CƠ SỞ DỮ LIỆU

6.1. Giới thiệu.

Như chúng ta đã được biết trong các bài học trước, ADO.NET là một tập các lớp nằm trong bộ thư viện lớp cơ sở của .NET Framework, cho phép các ứng dụng Windows (như C#, VB.NET) hay ứng dụng web (như ASP.NET) thao tác dễ dàng với các nguồn dữ liệu.

Sơ đồ liên kết quản lý cơ sở dữ liệu:



Trong đó:

- Tầng ứng dụng: bao gồm những giao diện người sử dụng cũng như những công cụ đơn giản mà người lập trình có thể dùng để xử lý dữ liệu theo các bài toán.

- Tầng kết nối dữ liệu: bao gồm tập hợp các công cụ, phương thức để kết nối tới những dữ liệu cần làm việc trong CSDL. Ở đây, tầng kết nối bao gồm các lớp đối tượng thuộc thư viện System.Data.SqlClient.
- Tầng cơ sở dữ liệu: bao gồm các bảng, các query trong cơ sở dữ liệu thực tại.

Như vậy để lập trình trên một CSDL phải sử dụng các đối tượng, các phương thức ở tầng kết nối như là những công cụ để có thể truy cập được vào CSDL tác nghiệp xử lý.

6.1.1. Kết nối ứng dụng với CSDL.

Khi xây dựng ứng dụng có kết nối là thao tác với dữ liệu trong các cơ sở dữ liệu khác nhau, thông thường ta sử dụng các bước sau:

- Bước 1: Xác định chuỗi kết nối và câu lệnh SQL cần thực hiện:

- Kết nối qua tài khoản Sql Server:

```
connStr1 = @"Data Source=<Server_chứa_CSDL>;  
Initial Catalog=<Tên_CSDL>;  
User ID=<Tên_truy_nhập>;  
password=<Mật_khẩu>";
```

- Kết nối qua tài khoản Windows:

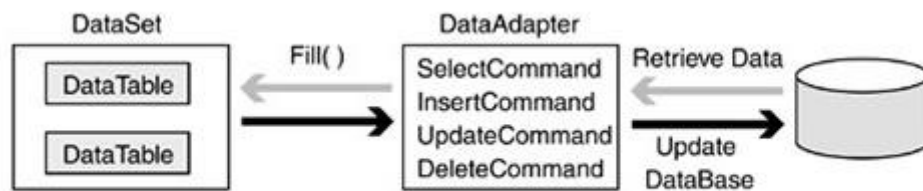
```
connStr2 = @"Data Source=<Server_chứa_CSDL>;  
Initial Catalog=<Tên_CSDL>;  
Integrated Security=True"
```

- Bước 2: tạo đối tượng connection kết nối giữa ứng dụng và CSDL

```
SqlConnection conn = new SqlConnection(); // khởi tạo đối tượng kết nối  
Conn.ConnectionString = connStr; // lấy đường dẫn đến cơ sở dữ liệu  
Conn.Open(); // mở kết nối
```

- Bước 3 : tạo đối tượng Adapter là cầu nối giữa dataset và datasource để thực hiện công việc như đọc hay cập nhật dữ liệu.

```
SqlDataAdapter <biến_SqlDataAdapter> = new SqlDataAdapter([Tham_số]) ;
```



- Bước 4: dữ liệu đọc ra từ câu lệnh select được lưu vào 1 DataTable trong DataSet

```
DataTable dt = new DataTable() ; // khởi tạo đối tượng datatable
```

```
da.Fill(dt) ; // đổ dữ liệu vào datatable
```

- Bước 5: Nếu muốn lọc dữ liệu theo một số tiêu chí nào đó, ta có thể sử dụng đối tượng DataView. DataView là lớp được dùng cho việc sắp xếp, lọc, tìm kiếm và thay đổi các bản ghi của một bảng:

```
DataView dv = new DataView(dt);
```

- Bước 6: hiển thị dữ liệu lên DataGridView.

Gán DataSource của dataGridView thông qua một DataTable (tên dt):

```
Dgr.DataSource = dt;
```

Gán DataSource của dataGridView thông qua một DataView (tên dv):

```
Dgr.DataSource = dv;
```

6.1.2. Kỹ thuật xử lý dữ liệu trên các điều khiển.

Một số kỹ thuật chính cần quan tâm:

- Sử dụng các điều khiển dữ liệu (ComboBox, DataGridView, ListView,...) để nhận nguồn dữ liệu và hiển thị với thuộc tính DataSource.
- Các thao tác: Đổ dữ liệu vào các điều khiển, nhận giá trị các ô (cell) khi người dùng bấm chuột chọn ô, gán giá trị các điều khiển cơ bản khác (TextBox, ListBox, ComboBox, CheckBox, RadioButton,...),....



6.2. Bài toán Quản lý kho hàng

Mô tả bài toán (Học viên có thể xem lại về yêu cầu bài toán quản lý nhập hàng ở bài số 1, trong đó có sẵn kịch bản SQL tạo database, tạo các bảng, các thủ tục lưu). Để tiện theo dõi, bài toán được mô tả vắn tắt lại như sau:

Một doanh nghiệp cần xây dựng hệ thống thông tin quản lý việc nhập hàng hóa trong kho hàng và một số các thống kê định kỳ. Các thông tin cần quản lý cho một mặt hàng bao gồm: mã hàng, tên hàng, đơn vị tính, thông tin về nhà cung cấp mặt hàng.

Khi có hàng hóa mới được nhập về, nhân viên quản lý kho sẽ lập một phiếu nhập kho. Trên phiếu nhập kho có ghi rõ các chi tiết: mã số phiếu, ngày lập phiếu, mã số nhà cung cấp, danh sách các mặt hàng nhập, hình thức thanh toán, thuế giá trị gia tăng và tổng số tiền mua hàng. Trên mỗi phiếu nhập có thể nhập nhiều thiết bị, ứng với từng thiết bị là số lượng thiết bị và đơn giá thiết bị. Ngoài ra, một số thông tin về nhà cung cấp cũng cần được lưu trữ như mã số nhà cung cấp, tên nhà cung cấp, địa chỉ, số điện thoại, mã số thuế, số tài khoản,....

Hệ thống quản lý xuất nhập hàng hóa sẽ cho phép người dùng :

- Cập nhật thông tin và các biến động thông tin về hàng hóa, nhà cung cấp,...
- Quản lý các phiếu nhập kho: lập phiếu, cập nhật phiếu.
- Thống kê báo cáo: lập bảng biểu thống kê số lượng hàng tồn kho hàng tháng, tổng số hàng nhập về trong tháng,...
- Tìm kiếm, tra cứu thông tin về tên hàng hóa, họ tên nhà cung cấp, mã số phiếu nhập, xuất, ngày nhập hàng,....

6.2.1. Phân tích và thiết kế

Việc phân tích và thiết kế bài toán, chúng ta cần đến kiến thức về phân tích và thiết kế hệ thống thông tin, kiến thức về cơ sở dữ liệu và kỹ thuật lập trình cơ sở dữ liệu. Trong phạm vi bài học, ta không đi vào phân tích và thiết kế hệ thống cũng như không trình bày cách xây dựng cơ sở dữ liệu cho bài toán. Ở đây, ta giả thiết cơ sở dữ liệu đã được triển khai với các thông tin kèm theo sau:

- Thông tin về cấu trúc cơ sở dữ liệu:

Các bảng dữ liệu: tblNhaCungcap, tblHang, tblHoadonNhap, tblChitietHoadonNhap.

- Thông tin về tài khoản người dùng:

- Server SQL: fithou
- Database: db_Kho
- Username: demo (vai trò sysadmin)
- Password: test

6.2.2. Lập trình ứng dụng

6.2.2.1. Các bước chuẩn bị

Công việc chuẩn bị cơ sở dữ liệu cho ứng dụng:

a) Phân cơ sở dữ liệu:

Tạo các bảng:

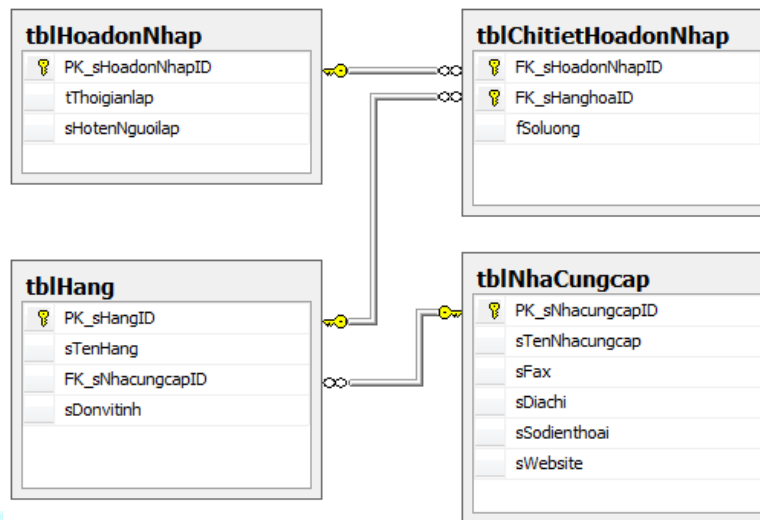
- Bảng tblNhaCungcap
- Bảng tblHang
- tblHoadonNhap
- tblChitietHoadonNhap

Nhập dữ liệu vào các bảng (mỗi bảng tối thiểu 5 bản ghi).

Tạo các thủ tục lưu (stored procedure) liên quan đến thao tác thêm, sửa, xóa dữ liệu trên các bảng.

Chú ý: Bạn có thể xem lại nội dung các lệnh tạo bảng, thủ tục lưu này trong tệp kịch bản CSDL quản lý kho ở bài 1.

Mô hình quan hệ giữa các bảng trong cơ sở dữ liệu quản lý kho:



Tạo các truy vấn (view) truy vấn dữ liệu trên các bảng.

- View vNhaCungcap:

```
Create View vNhaCungcap
AS
```

```
Select PK_sNhaCungcapID as [Mã NCC],
       sTenNhaCungcap as [Tên Nhà cung cấp],
       sFax as [Fax],
       sDiachi as [Địa chỉ],
       sSodienthoai as [Số điện thoại],
       sWebsite as [Website]
```

```
From tblNhaCungcapView
```

- vHang:

```
Create View vHang
AS
```

```
Select PK_sHangID as [Mã hàng],
       sTenHang as [Tên hàng],
       sTenNhaCungcap [Nhà cung cấp],
       sDonvitinh as [Đơn vị tính]
From tblHang a Inner join tblNhaCungcap b
On(a.FK_sNhaCungcapID=b.PK_sNhaCungcapID)
```

- View vHoadonNhap: Bạn đọc tham khảo và tự viết.

- View vChitietHoadon:

```
Create View vChitietHoadon
AS
Select
    PK_sHoadonnhapID as [Số HĐ],
    tThoigianlap as [Ngày lập],
    sHotenNguoilap as [Người lập],
    sTenNhacongcap as [Nhà cung cấp],
    sTenHang as [Tên hàng],
    fSoluong as [Số lượng]
From tblHang a Inner join tblChitietHoadonNhap b
    ON a.PK_sHangID = b.FK_sHanghoaID
    Inner Join tblNhaCungcap c
    On a.FK_sNhacongcapID = c.PK_sNhacongcapID
    Inner join tblHoadonNhap d
    On b.FK_sHoadonNhapID = d.PK_sHoadonNhapID
```

b) Phần ứng dụng winform:

Sau khi hoàn thành bước chuẩn bị cơ sở dữ liệu cho ứng dụng, ta thiết kế các form chương trình:

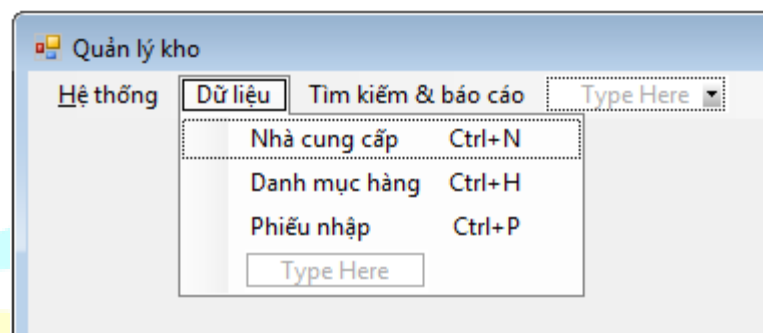
- Form chính (frmMain): chứa các menu chức năng gọi hiển thị các form con thực thi các yêu cầu của bài toán.
- Các form cập nhật dữ liệu: thêm, sửa, xóa dữ liệu trên các bảng.
- Các form tìm kiếm dữ liệu theo các tiêu chí nào đó như: tìm kiếm theo nhóm hàng, tên hàng, nhà cung cấp; tìm kiếm thông tin chi tiết các phiếu nhập kho theo ngày nhập (hoặc khoảng thời gian nhập), theo nhà cung cấp,...
- Các form hiển thị báo cáo liên quan.

6.2.2.2. Các bước triển khai ứng dụng

a) Xây dựng form giao diện chính

Các bước thực hiện:

- Bước 1: Khởi động Visual Studio và tạo một project mới, đặt tên project là “QuanlyKho”.
- Bước 2: Đặt tên (frmMain) và tiêu đề form chính (“Chương trình quản lý kho”)
- Bước 3: Thêm vào form 1 điều khiển MenuStrip, thêm các menu con như hình vẽ.

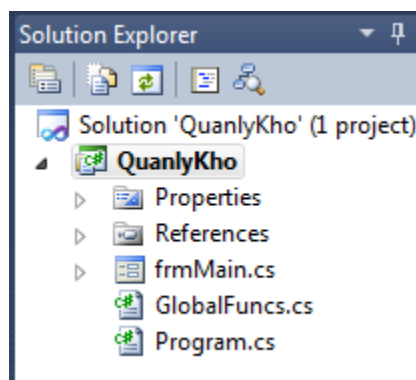


b) Xây dựng lớp chứa các biến và hàm dùng chung.

Trong quá trình triển khai mã lệnh, sẽ có nhiều đoạn lệnh trùng lặp về ý tưởng và cả về nội dung. Nếu tại mỗi form ta đều viết lại các đoạn lệnh này thì dư thừa nhiều đoạn mã lệnh cũng như khó sửa lỗi phát sinh về sau (nếu có). Vì thế, trước khi xây dựng các form giao diện, ta xây dựng một lớp chung chứa các biến, hàm dùng chung. Thông qua các phương thức của lớp này, ta sẽ thao tác với cơ sở dữ liệu bằng các gọi thực thi các phương thức của lớp với các tham số truyền vào là khác nhau tùy vào mã lệnh được đặt ở mỗi form trong chương trình.

- Các bước thực hiện:

- Bước 1: Thêm một class mới vào project, đặt tên lớp là GlobalFuncs



- Bước 2: Khai báo các thư viện và biến chung:

- Khai báo thư viện:

```
using System.Data;  
using System.Data.SqlClient;  
using System.Windows.Forms;
```

- Khai báo chuỗi kết nối và biến SqlConnection

```
public string strCon = @"Server=fithou; Database=QLKho;  
                        User  
                        ID=demo;Password=test";  
public SqlConnection myCnn = new SqlConnection();
```

- Bước 3: Xây dựng các phương thức:

- Phương thức kết nối cơ sở dữ liệu:

Hàm kết nối cơ sở dữ liệu

```
public bool KetnoiCSDL()  
{  
    try  
    {  
        if (myCnn.State == ConnectionState.Open)  
            myCnn.Close();  
        myCnn.ConnectionString = strCon;  
        myCnn.Open();  
    }  
    catch  
    {  
        MessageBox.Show("Lỗi kết nối cơ sở dữ liệu", "Thông báo");  
        return false;  
    }  
    return true;  
}
```

- Phương thức lấy dữ liệu từ một view và đổ vào một dataGridView được truyền vào từ nơi gọi:

Lấy dữ liệu đưa vào DataGridView

```
public void HienthiDulieutrenDatagridView(string strSQL, DataGridView
dgr)
{ if (KetnoiCSDL() == false)
    return;
try
{ SqlDataAdapter da = new SqlDataAdapter("Select * from " + strSQL, myCnn);
  DataTable dt = new DataTable();
  da.Fill(dt);
  dgr.DataSource = dt;
  dgr.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;
  da.Dispose();
  dt.Dispose();
}
catch
{ MessageBox.Show("Lỗi hiển thị dữ liệu", "Thông báo");
  return;
}
}
```

- Các phương thức kiểm tra dữ liệu kiểu số nguyên, thực, ngày tháng nhập vào có hợp lệ hay không. Kết quả trả về đều là “true” nếu hợp lệ:

```
//Kiểm tra giá trị số thực
public bool KiemtraDulieuSothuc(string strSo)
{ try
  { float.Parse(strSo);
  }
  catch
  { return false;
  }
  return true;
}
//Kiểm tra giá trị số nguyên
```

```
public bool KiemtraDulieuSonguyen(string strSo)
{
    try
    {
        int.Parse(strSo);
    }
    catch
    {
        return false;
    }
    return true;
}

//Kiểm tra dữ liệu ngày tháng
public bool KiemtraDulieuNgaythang(string strNgaythang)
{
    try
    {
        DateTime.Parse(strNgaythang);
    }
    catch
    {
        return false;
    }
    return true;
}
```

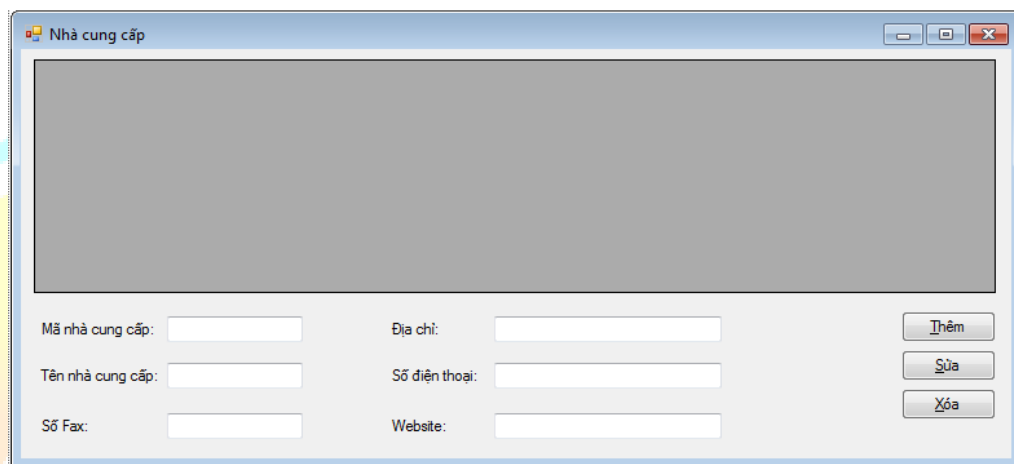
- Phương thức xóa dữ liệu trong bảng, tham số truyền vào là các thông tin về tên thủ tục lưu (đã tạo trước), tên tham số và giá trị tham số. Thủ tục này dùng để xóa các bản ghi trong các bảng có khóa chính là 1 trường. Bạn có thể tạo thêm các thủ tục xóa khác để áp dụng cho các trường hợp khóa chính là kết hợp của nhiều trường trong bảng:

```
//Xóa dữ liệu trong bảng
public void Xoadulieu(string sTenThutuc, string sThamso, string sGiatri)
{
    if (KetnoiCSDL() == false)
        return;
    try
    {
        SqlCommand cmd = new SqlCommand(sTenThutuc, myCnn);
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.Parameters.AddWithValue(sThamso, sGiatri);
        cmd.ExecuteNonQuery();
    }
}
```

```
}  
catch  
{ MessageBox.Show("Lỗi xóa dữ liệu", "Thông báo");  
}  
}
```

c) Xây dựng form cập nhật thông tin nhà cung cấp

Giao diện thiết kế:



Diễn giải các thao tác:

- Khi form hiển thị, danh sách nhà cung cấp sẽ được hiển thị trên dgrNhaCC.
- Sau khi người dùng nhập dữ liệu vào các điều khiển mã nhà cung cấp, tên nhà cung cấp,... và bấm nút btnThem, chương trình sẽ kiểm tra tính hợp lệ của dữ liệu nhập (hợp lệ về kiểu dữ liệu, miền giá trị, kiểm tra trùng khóa chính), nếu hợp lệ sẽ ghi dữ liệu nhập của người dùng vào bảng.
- Người dùng muốn sửa dữ liệu nhà cung cấp nào thì nhấp chuột vào bản ghi tương ứng trên dgrNhaCC, thông tin về nhà cung cấp đó sẽ được đưa vào các điều khiển. Sau khi tiến hành thao tác sửa dữ liệu và bấm nút btnSua, chương trình sẽ kiểm tra tính hợp lệ của dữ liệu nhập (hợp lệ về kiểu dữ liệu, miền giá trị, kiểm tra trùng khóa chính nếu người dùng có sửa khóa chính), nếu hợp lệ sẽ cập nhật dữ liệu mới của người dùng vào bảng.

- Người dùng muốn xóa dữ liệu nhà cung cấp nào thì nhấp chuột vào bản ghi tương ứng trên dgrNhaCC, thông tin về nhà cung cấp đó sẽ được đưa vào các điều khiển. Sau khi bấm nút btnXoa, chương trình có hỏi xác nhận thao tác và nếu người dùng chọn đồng ý (“Yes”) thì bản ghi tương ứng sẽ được xóa khỏi bảng.

Các bước thực hiện:

- Bước 1: Thêm vào project một form mới (đặt tên frmNhaCungcap). Trên form ta đưa vào các điều khiển sau:

Điều khiển	Tên
datagridView	dgrNhaCC
textBox	txtMaNhaCC
textBox	txtTenNhaCC
textBox	txtFax
textBox	txtDiachi
textBox	txtSoDienthoai
textBox	txtWebsite
button	btnThem
button	btnSua
button	btnXoa
Các label lời nhắc	

- Bước 2: Viết lệnh khai báo biến và xử lý các sự kiện:

- Khai báo thư viện và các biến:

- Khai báo thư viện:

```
using System.Data.SqlClient;
```

- Khai báo biến kiểu lớp GlobalFuncs để thực thi các phương thức liên quan và một chuỗi lưu tên view dùng để hiển thị dữ liệu trên dgrNhaCC:

```
GlobalFuncs gf = new GlobalFuncs();  
string strNhaCC = "vNhaCungcap";
```

- Sự kiện form_load: gọi hàm hiển thị thông tin nhà cung cấp (lấy từ vNhaCungcap)

```
gf.HienthiDulieutrenDataGridView(strNhaCC, dgrNhaCC);
```

- Sự kiện dgrNhaCC_Click:

```
private void dgrNhaCC_Click(object sender, EventArgs e)  
{  
    if (dgrNhaCC.Rows.Count <= 0)  
        return;  
    txtMaNhaCC.Text = dgrNhaCC.CurrentRow.Cells[0].Value.ToString();  
    txtTenNhaCC.Text = dgrNhaCC.CurrentRow.Cells[1].Value.ToString();  
    txtFax.Text = dgrNhaCC.CurrentRow.Cells[2].Value.ToString();  
    txtDiachi.Text = dgrNhaCC.CurrentRow.Cells[3].Value.ToString();  
    txtSoDienthoai.Text = dgrNhaCC.CurrentRow.Cells[4].Value.ToString();  
    txtWebsite.Text = dgrNhaCC.CurrentRow.Cells[5].Value.ToString();  
}
```

- Sự kiện btnThem_Click:

```
private void btnThem_Click(object sender, EventArgs e)  
{  
    //Các bước cần thực hiện:  
    //1. Kiểm tra tính hợp lệ của dữ liệu  
    if (txtMaNhaCC.Text.Trim().Length == 0 || txtTenNhaCC.Text.Trim().Length == 0)  
    {  
        MessageBox.Show("Chưa nhập đủ mã/tên nhà cung cấp!", "Thông báo");  
        return;  
    }  
    //2. Kiểm tra trùng khóa chính/Unique  
    if (gf.KiemtraTrungkhoa("tblNhaCungcap", "PK_sNhaCungcapID",  
                           txtMaNhaCC.Text) == true)  
    {  
        MessageBox.Show("Mã nhà cung cấp này đã có!", "Thông báo");  
        return;  
    }  
    //Kiểm tra trùng tên nhà cung cấp?
```

```
if (gf.KiemtraTrungkhoa("tblNhaCungcap", "sTenNhacungcap",  
                        txtTenNhaCC.Text) == true)  
{ MessageBox.Show("Tên nhà cung cấp này đã có!", "Thông báo");  
  return;  
}  
  
//3. Thêm dữ liệu  
try  
{ if (gf.KetnoiCSDL() == false)  
    return;  
  SqlCommand cmd = new SqlCommand("spNhaCungcap_Insert", gf.myCnn);  
  cmd.CommandType = CommandType.StoredProcedure;  
  cmd.Parameters.AddWithValue("@PK_sNhacungcapID", txtMaNhaCC.Text);  
  cmd.Parameters.AddWithValue("@sTenNhacungcap",  
txtTenNhaCC.Text);  
  cmd.Parameters.AddWithValue("@sFax", txtFax.Text);  
  cmd.Parameters.AddWithValue("@sDiachi", txtDiachi.Text);  
  cmd.Parameters.AddWithValue("@sSodienthoai", txtSoDienthoai.Text);  
  cmd.Parameters.AddWithValue("@sWebsite", txtWebsite.Text);  
  cmd.ExecuteNonQuery();  
  gf.HienthiDulieutrenDatagridView(strNhaCC, dgrNhaCC);  
  cmd.Dispose();  
}  
catch  
{ MessageBox.Show("Lỗi thêm dữ liệu", "Thông báo");  
}  
}
```

- Sự kiện btnSua_Click:

```
private void btnSua_Click(object sender, EventArgs e)  
{ //Các bước thực hiện:  
  //1. Kiểm soát không cho sửa mã nhà cung cấp  
  if (txtMaNhaCC.Text != dgrNhaCC.CurrentRow.Cells[0].Value.ToString())  
  { MessageBox.Show("Không thể sửa mã nhà cung cấp");  
    return;  
  }
```



```
}  
  
//2. Kiểm tra tính hợp lệ của dữ liệu  
if (txtMaNhaCC.Text.Trim().Length == 0 || txtTenNhaCC.Text.Trim().Length == 0)  
{ MessageBox.Show("Chưa nhập đủ mã/tên nhà cung cấp!", "Thông báo");  
  return;  
}  
  
//3. Kiểm tra trùng khóa chính/Unique  
if (txtTenNhaCC.Text != dgrNhaCC.CurrentRow.Cells[1].Value.ToString())  
{ //Kiểm tra trùng tên nhà cung cấp?  
  if (gf.KiemtraTrungkhoa("tblNhaCungcap", "sTenNhaCungcap",  
                           txtTenNhaCC.Text) == true)  
  { MessageBox.Show("Tên nhà cung cấp này đã có!", "Thông báo");  
    return;  
  }  
}  
  
//4. Sửa dữ liệu  
try  
{ if (gf.KetnoiCSDL() == false)  
  return;  
  SqlCommand cmd = new SqlCommand("spNhaCungcap_UpdateByPK", gf.myCnn);  
  cmd.CommandType = CommandType.StoredProcedure;  
  cmd.Parameters.AddWithValue("@PK_sNhaCungcapID",  
                             txtMaNhaCC.Text);  
  cmd.Parameters.AddWithValue("@sTenNhaCungcap", txtTenNhaCC.Text);  
  cmd.Parameters.AddWithValue("@sFax", txtFax.Text);  
  cmd.Parameters.AddWithValue("@sDiachi", txtDiachi.Text);  
  cmd.Parameters.AddWithValue("@sSoDienthoai",  
                             txtSoDienthoai.Text);  
  cmd.Parameters.AddWithValue("@sWebsite", txtWebsite.Text);  
  cmd.ExecuteNonQuery();  
  gf.HienthiDulieutrenDataGridView(strNhaCC, dgrNhaCC);  
  cmd.Dispose();  
}  
catch
```

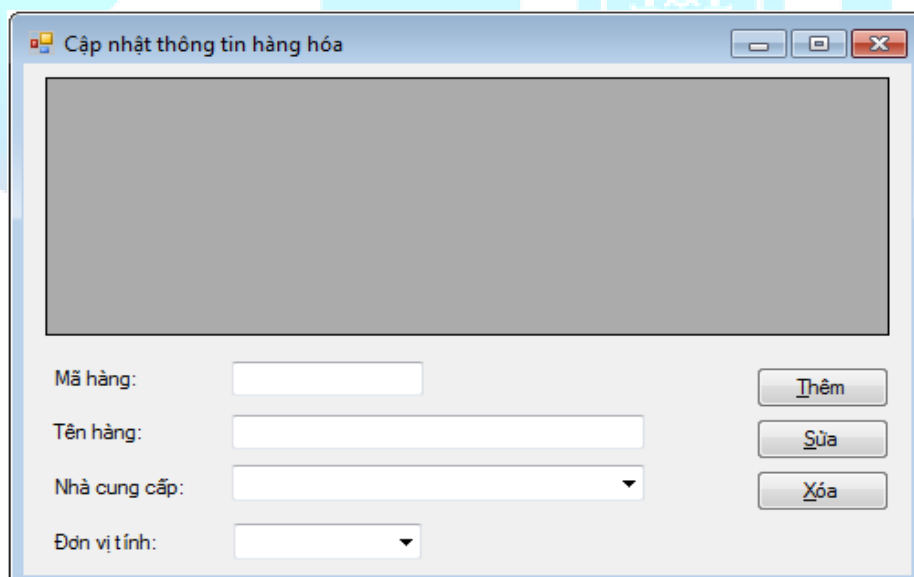
```
{ MessageBox.Show("Lỗi sửa dữ liệu", "Thông báo");  
}  
}
```

- Sự kiện btnXoa_Click:

```
private void btnXoa_Click(object sender, EventArgs e)  
{ if (dgrNhaCC.Rows.Count <= 0)  
    return;  
    if (MessageBox.Show("Xác nhận xóa nhà cung cấp?", "Thông báo",  
        MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==  
        DialogResult.No)  
        return;  
    gf.Xoadulieu("spNhaCungcap_DeleteByPK", "@PK_sNhaCungcapID",  
        dgrNhaCC.CurrentRow.Cells[0].Value.ToString());  
    gf.HienthiDulieutrenDatagridView(strNhaCC, dgrNhaCC);  
}
```

c) Xây dựng form cập nhật thông tin các mặt hàng trong kho

Giao diện thiết kế:



Diễn giải các thao tác:

- Khi form hiển thị, danh sách các mặt hàng trong kho sẽ được hiển thị trên dgrHang.
- Sau khi người dùng nhập dữ liệu vào các điều khiển mã hàng, tên hàng, tên nhà cung cấp (chọn từ cboNhaCC), đơn vị tính (chọn từ cboDonvitinh) và bấm nút btnThem, chương trình sẽ kiểm tra tính hợp lệ của dữ liệu nhập (hợp lệ về kiểu dữ liệu, miền giá trị, kiểm tra trùng khóa chính), nếu hợp lệ sẽ ghi dữ liệu nhập của người dùng vào bảng.
- Người dùng muốn sửa dữ liệu mặt hàng nào thì nhấp chuột vào bản ghi tương ứng trên dgrHang, thông tin về mặt hàng đó sẽ được đưa vào các điều khiển. Sau khi tiến hành thao tác sửa dữ liệu và bấm nút btnSua, chương trình sẽ kiểm tra tính hợp lệ của dữ liệu nhập (hợp lệ về kiểu dữ liệu, miền giá trị, kiểm tra trùng khóa chính nếu người dùng có sửa khóa chính), nếu hợp lệ sẽ cập nhật dữ liệu mới của người dùng vào bảng.
- Người dùng muốn xóa dữ liệu mặt hàng nào thì nhấp chuột vào bản ghi tương ứng trên dgrHang, thông tin về mặt hàng đó sẽ được đưa vào các điều khiển. Sau khi bấm nút btnXoa, chương trình có hỏi xác nhận thao tác và nếu người dùng chọn đồng ý (“Yes”) thì bản ghi tương ứng sẽ được xóa khỏi bảng.

Các bước thực hiện:

- Bước 1: Thêm vào project một form mới (đặt tên frmHang). Trên form ta đưa vào các điều khiển sau:

Điều khiển	Tên
datagridView	dgrHang
textBox	txtMahang
textBox	txtTenhang
comboBox	cboNhaCC
comboBox	cboDonvitinh
button	btnThem

button	btnSua
button	btnXoa
Các label lời nhắc	

- Trên form ta thiết kế 2 comboBox hiển thị thông tin nhà cung cấp và đơn vị tính để hỗ trợ người dùng chọn lựa thông tin nhanh chóng, thêm nữa, khi nhập thông tin một mặt hàng mới thì phải đảm bảo mặt hàng đó được cung cấp bởi một nhà cung cấp cụ thể có trong cboNhaCC, điều này đảm bảo tính toàn vẹn của dữ liệu nhập. Để nạp thông tin nhà cung cấp vào cboNhaCC, ta viết thêm hàm HienthiDulieutrenComboBox trong lớp GlobalFuncs với các tham số truyền vào là tên của bảng dữ liệu, tên của các trường hiển thị (displayMember), tên của trường liên kết (ValueMember), điều khiển comboBox để nhận dữ liệu. Nội dung hàm cụ thể như sau:

```
//Bổ sung hàm HienthiDulieutrenComboBox trong lớp GlobalFuncs
//Lấy dữ liệu từ bảng và đưa vào ComboBox
public void HienthiDulieutrenComboBox(string sTenbang, string
                                     sTenCotKhoa, string sTenCotHienthi, ComboBox cbo)
{ if (KetnoiCSDL() == false)
    return;
    string strSQL="Select "+sTenCotKhoa+", "+sTenCotHienthi+" From
"+sTenbang;
    SqlDataAdapter da = new SqlDataAdapter(strSQL, myCnn);
    DataTable dt = new DataTable();
    da.Fill(dt);
    cbo.DataSource = dt;
    cbo.DisplayMember = sTenCotHienthi;
    cbo.ValueMember = sTenCotKhoa;
    da.Dispose();
}
```

- Bước 2: Viết lệnh khai báo biến và xử lý các sự kiện:
 - Khai báo thư viện và các biến:

- Khai báo thư viện:

```
using System.Data.SqlClient;
```

- Khai báo biến kiểu lớp GlobalFuncs để thực thi các phương thức liên quan và một chuỗi lưu tên view dùng để hiển thị dữ liệu trên dgvNhaCC:

```
GlobalFuncs gf = new GlobalFuncs();  
string strNhaCC = "vHang";
```

- Sự kiện form_load: gọi hàm hiển thị thông tin nhà cung cấp (lấy từ vNhaCungcap), gọi hàm nạp dữ liệu nhà cung cấp vào cboNhaCC và nạp trực tiếp một số đơn vị tính thường dùng vào cboDonvitinh:

```
gf.HienthiDulieutrenDataGridView(strHang, dgvHang);  
gf.HienthiDulieutrenComboBox("tblNhaCungcap",  
"PK_sNhaCungcapID",  
"sTenNhaCungcap", cboNhaCC);  
cboDonvitinh.Items.Add("Chiếc");  
cboDonvitinh.Items.Add("Bộ");  
cboDonvitinh.Items.Add("Thanh");  
cboDonvitinh.SelectedIndex = 0;
```

- Sự kiện dgvHang_Click:

```
private void dgvHang_Click(object sender, EventArgs e)  
{ if (dgvHang.Rows.Count <= 0)  
    return;  
    txtMahang.Text = dgvHang.CurrentRow.Cells[0].Value.ToString();  
    txtTenhang.Text = dgvHang.CurrentRow.Cells[1].Value.ToString();  
    cboNhaCC.Text = dgvHang.CurrentRow.Cells[2].Value.ToString();  
    cboDonvitinh.Text = dgvHang.CurrentRow.Cells[3].Value.ToString();  
}
```

- Sự kiện btnThem_Click:

```
private void btnThem_Click(object sender, EventArgs e)  
{ //Các bước cần thực hiện:  
    //1. Kiểm tra tính hợp lệ của dữ liệu
```

```
if (txtMahang.Text.Trim().Length == 0 || txtTenhang.Text.Trim().Length == 0)
{
    MessageBox.Show("Chưa nhập đủ mã/tên hàng!", "Thông báo");
    return;
}

//2. Kiểm tra trùng khóa chính/Unique
if (gf.KiemtraTrungkhoa("tblhang", "PK_sHangID", txtMahang.Text) == true)
{
    MessageBox.Show("Mã hàng này đã có!", "Thông báo");
    return;
}

//Kiểm tra trùng tên hàng?
if (gf.KiemtraTrungkhoa("tblhang", "sTenHang", txtTenhang.Text) == true)
{
    MessageBox.Show("Tên hàng này đã có!", "Thông báo");
    return;
}

//3. Thêm dữ liệu
try
{
    if (gf.KetnoiCSDL() == false)
        return;

    SqlCommand cmd = new SqlCommand("spHang_Insert", gf.myCnn);
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Parameters.AddWithValue("@PK_sHangID", txtMahang.Text);
    cmd.Parameters.AddWithValue("@sTenHang", txtTenhang.Text);
    cmd.Parameters.AddWithValue("@FK_sNhacongcapID",
        cboNhaCC.SelectedValue.ToString());
    cmd.Parameters.AddWithValue("@sDonvitinh", cboDonvitinh.Text);
    cmd.ExecuteNonQuery();
    gf.HienthiDulieutrenDataGridView(strHang, dgrHang);
    cmd.Dispose();
}
catch
{
    MessageBox.Show("Lỗi thêm dữ liệu", "Thông báo");
}
}
```

- Sự kiện btnSua_Click:

```
private void btnSua_Click(object sender, EventArgs e)
{ //Các bước thực hiện:
  //1. Kiểm soát không cho sửa mã hàng
  if (txtMahang.Text != dgrHang.CurrentRow.Cells[0].Value.ToString())
  { MessageBox.Show("Không thể sửa mã hàng");
    return;
  }
  //2. Kiểm tra tính hợp lệ của dữ liệu
  if (txtMahang.Text.Trim().Length == 0 || txtTenhang.Text.Trim().Length == 0)
  { MessageBox.Show("Chưa nhập đủ mã/tên hàng!", "Thông báo");
    return;
  }
  //3. Kiểm tra trùng khóa chính/Unique
  if (txtTenhang.Text != dgrHang.CurrentRow.Cells[1].Value.ToString())
  { //Kiểm tra trùng tên hàng?
    if (gf.KiemtraTrungkhoa("tblHang", "sTenHang", txtTenhang.Text) == true)
    { MessageBox.Show("Tên mặt hàng này đã có!", "Thông báo");
      return;
    }
  }
  //4. Sửa dữ liệu
  try
  { if (gf.KetnoiCSDL() == false)
    { return;
      SqlCommand cmd = new SqlCommand("spHang_UpdateByPK",
gf.myCnn);
      cmd.CommandType = CommandType.StoredProcedure;
      cmd.Parameters.AddWithValue("@PK_sHangID", txtMahang.Text);
      cmd.Parameters.AddWithValue("@sTenHang", txtTenhang.Text);
      cmd.Parameters.AddWithValue("@FK_sNhacongcapID",
cboNhaCC.SelectedValue.ToString());
      cmd.Parameters.AddWithValue("@sDonvitinh", cboDonvitinh.Text);
    }
  }
}
```

```
cmd.ExecuteNonQuery();  
gf.HienthiDulieutrenDatagridView(strHang, dgrHang);  
cmd.Dispose();  
}  
catch  
{ MessageBox.Show("Lỗi sửa dữ liệu", "Thông báo");  
}  
}
```

- Sự kiện btnXoa_Click:

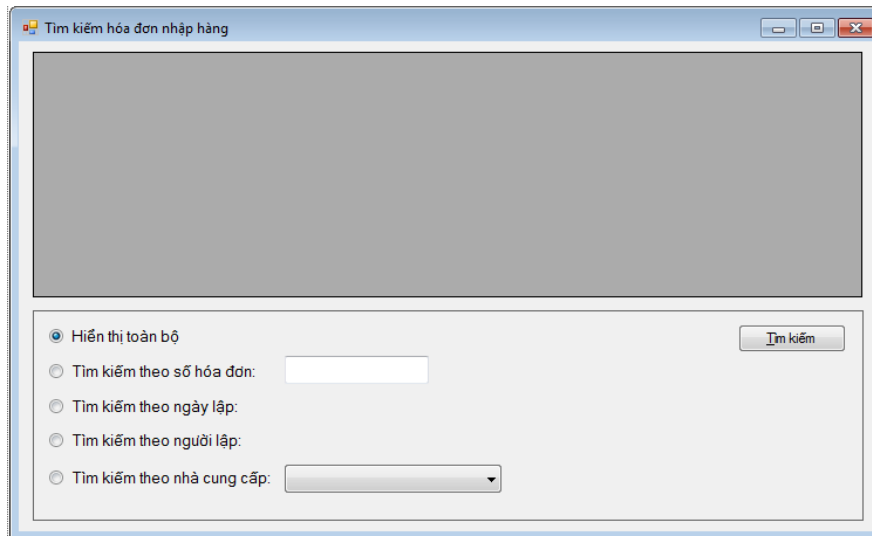
```
private void btnXoa_Click(object sender, EventArgs e)  
{ if (dgrHang.Rows.Count <= 0)  
    return;  
    if (MessageBox.Show("Xác nhận xóa mặt hàng?", "Thông báo",  
        MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==  
        DialogResult.No)  
        return;  
    gf.Xoadulieu("spHang_DeleteByPK", "@PK_sHangID",  
        dgrHang.CurrentRow.Cells[0].Value.ToString());  
    gf.HienthiDulieutrenDatagridView(strHang, dgrHang);  
}
```

d) Xây dựng form cập nhật thông tin các phiếu nhập hàng

Bạn đọc tham khảo từ 2 mẫu xây dựng form trên để thiết kế form nhập thông tin các phiếu nhập hàng cũng như cập nhật chi tiết các phiếu hàng.

e) Xây dựng form tìm kiếm thông tin các mặt hàng

Mẫu form thiết kế:



Diễn giải các thao tác:

- Khi form hiển thị, danh sách chi tiết các hóa đơn nhập hàng sẽ được hiển thị trên form.
- Người dùng sẽ chọn một trong các tiêu chí tìm kiếm (hiển thị toàn, tìm theo số hóa đơn, tìm theo thời gian lập hóa đơn, tìm theo tên người lập hóa đơn, tìm theo nhà cung cấp hàng), sau đó nhập giá trị cần tìm vào textBox txtDieukien.
- Khi người dùng bấm nút “Tìm kiếm”, chương trình sẽ thực hiện câu lệnh truy vấn theo tiêu chí được người dùng chọn và hiển thị kết quả trên dataGridView.

Các bước thực hiện:

- Bước 1: Thêm vào project một form mới (đặt tên frmTimkiemHoadon).

Trên form ta đưa vào các điều khiển sau:

Điều khiển	Tên
dataGridView	dgrHoadon
radioButton	rbtToanbo
radioButton	rbtSoHD
radioButton	rbtNgaylap
radioButton	rbtNguoilap
radioButton	rbtNhacungcap

radioButton	cboNhacungcap
textBox	txtDieukien
button	btnTimkiem

- Bước 2: Viết lệnh khai báo biến và xử lý các sự kiện:
 - Khai báo biến kiểu lớp GlobalFuncs để thực thi các phương thức liên quan và một chuỗi lưu tên view dùng để hiển thị dữ liệu trên dgrHoadon:

```
GlobalFuncs gf = new GlobalFuncs();  
string strSelect = "vChitietHoadon";
```

- Viết hàm Andieukhien (ẩn điều khiển) phục vụ việc ẩn/ hiển thị các điều khiển để hỗ trợ người dùng dễ thao tác với ứng dụng.

```
private void AnDieukhien()  
{ txtDieukien.Visible = false;  
  cboNhacungcap.Visible = false;  
}
```

- Sự kiện form_load: gọi hàm hiển thị thông tin chi tiết các hóa đơn nhập hàng (lấy từ vChitietHoadon) và gọi hàm nạp dữ liệu nhà cung cấp vào cboNhaCC:

```
private void frmTimkiemHoadon_Load(object sender, EventArgs e)  
{ AnDieukhien();  
  gf.HienthiDulieutrenDataGridView(strSelect, dgrHoadon);  
  gf.HienthiDulieutrenComboBox("tblNhacungcap",  
    "PK_sNhacungcapID",  
    "sTenNhacungcap", cboNhacungcap);  
}
```

- Các sự kiện Checked_Change của các radioButton: ta gọi hàm AnDieukhien, gán vị trí mới cho textbox txtDieukien (hoặc comboBox cboNhacungcap), ví dụ với sự kiện rbtNgaylap_CheckedChanged (các sự kiện của các nút chọn khác bạn đọc tham khảo và tự viết):

```
private void rbtNgaylap_CheckedChanged(object sender, EventArgs e)
```

```
{ AnDieukhien();  
  txtDieukien.Visible = true;  
  txtDieukien.Top = rbtNgaylap.Top;  
}
```

- Sự kiện btnTimkiem_Click:

```
private void btnTimkiem_Click(object sender, EventArgs e)  
{ string s = strSelect;  
  if (rbtSoHD.Checked)  
  { s += " WHERE [Số HĐ]=''" + txtDieukien.Text + "'";  
  }  
  if (rbtNgaylap.Checked)  
  { if(!gf.KiemtraDulieuNgaythang(txtDieukien.Text))  
    { MessageBox.Show("Nhập sai ngày tháng");  
      return;  
    }  
    s += " WHERE convert(Varchar(10),[Ngày lập],103)='" + txtDieukien.Text +  
    "'";  
  }  
  if (rbtNguoilap.Checked)  
  { s += " WHERE [Người lập] like N'%" + txtDieukien.Text + "'";  
  }  
  if (rbtNhacungcap.Checked)  
  { s += " WHERE [Nhà cung cấp]=N'" + cboNhacungcap.Text + "'";  
  }  
  gf.HienthiDulieutrenDataGridView(s, dgrHoadon);  
}
```

Giao diện một số kết quả tìm kiếm:

- Hiển thị toàn bộ danh sách:

The screenshot shows a window titled "Tìm kiếm hóa đơn nhập hàng". It contains a table with the following data:

Số HD	Ngày lập	Người lập	Nhà cung cấp	Tên hàng	Số lượng
HD001	21/11/2015	Trần Văn Hùng	Công ty Thịnh Hưng	Ổ cứng Seagate 3...	1
HD001	21/11/2015	Trần Văn Hùng	Á Châu Company	USB Kingston 16Gb	3
HD002	21/03/2015	Lê Thu Hà	Á Châu Company	Ram 8Gb Kingston	10
HD002	21/03/2015	Lê Thu Hà	Công ty TNHH Sa...	Speaker 4.1 MAX	15
HD002	21/03/2015	Lê Thu Hà	Á Châu Company	USB Kingston 16Gb	8
HD003	01/03/2015	Nguyễn Hồng Quân	Công ty Thịnh Hưng	Ổ cứng Seagate 5...	5
HD004	12/04/2015	Trần Văn Hùng	Công ty Thịnh Hưng	Ổ cứng Seagate 3...	10
HD005	18/10/2015	Lê Thu Hà	Á Châu Company	USB Kingston 16Gb	17

Below the table, there are radio buttons for search criteria: "Hiện thị toàn bộ" (selected), "Tìm kiếm theo số hóa đơn:", "Tìm kiếm theo ngày lập:", "Tìm kiếm theo người lập:", and "Tìm kiếm theo nhà cung cấp:". A "Tìm kiếm" button is also present.

- Hiển thị chi tiết hóa đơn nhập theo ngày cụ thể “21/03/2015”:

The screenshot shows the same window as before, but with the search criteria set to "Tìm kiếm theo ngày lập:" and the date "21/03/2015" entered in the text box. The table now displays only the invoices from that date:

Số HD	Ngày lập	Người lập	Nhà cung cấp	Tên hàng	Số lượng
HD002	21/03/2015	Lê Thu Hà	Á Châu Company	Ram 8Gb Kingston	10
HD002	21/03/2015	Lê Thu Hà	Công ty TNHH Sao...	Speaker 4.1 MAX	15
HD002	21/03/2015	Lê Thu Hà	Á Châu Company	USB Kingston 16Gb	8

The search criteria section shows "Hiện thị toàn bộ" as selected, but the date filter is active. The "Tìm kiếm" button is highlighted.

Chúc Anh/Chị học tập tốt!